Operating System Assignment

Name:- Sayan Mondal Dept.:- CSE
SEM: 5 Roll: 33200118015

1. What do you mean by Operating System? What are functions of an operating system?

A:- An operating system is an interface between a computer user and computer hardware which performs all the basic tasks like file management memory management process management handling output and input and controlling peripheral devices such as disk drives and printers.

• Functions of OS:

An operating system is an interface between computer's hardware and user himself whose functions are;

i) Manage the computer resources such as:
   - a) CPU
   - b) Memory
   - c) Disk Drives & printers

ii) Establish a user interface (UI);

iii) Execute and provide services for application software.

Sayan Mondal Roll-15

2. What do you mean by Multitasking?

A: Multitasking is the process of running more than 1 program in one computer at the same time. It is used to keep all of a computer's resources at work as much of time as possible.

In terms of OS; Allowing a user to perform more than one computer task at a time. The OS is able to keep track of where you are in these tasks and go from one to ~~other~~ without losing information.

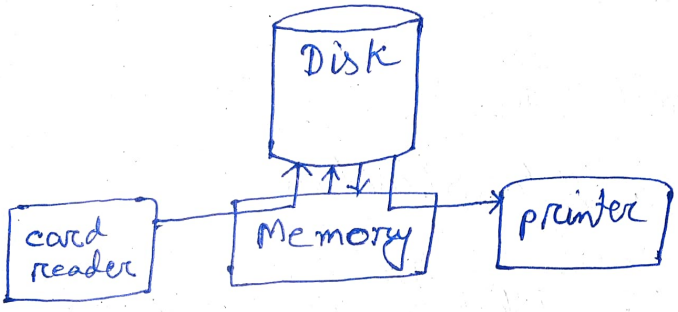
MS windows 2000, IBM OS/390 are some examples of multitasking OS's

3. What do you mean by Multiprogramming?

A: It is the ability of an OS to execute more than one task/program at a time. It is used to keep all of ~~the~~ a computer's resources at work as much of the time as possible on a single processor machine. More than one task/program/job/process can reside into the main memory at one point of time. A computer running reside into the main memory at one point of time. A computer running excel and firefox browser simultaneously is an example of multiprogra-mming.

4. What do you mean SPOOLING?

A: Simultaneously peripheral operations online refers to putting data of various I/O jobs in buffer. (It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing it's computing task). This buffer is a special area in memory or hard disk which is accessible to I/O drives.

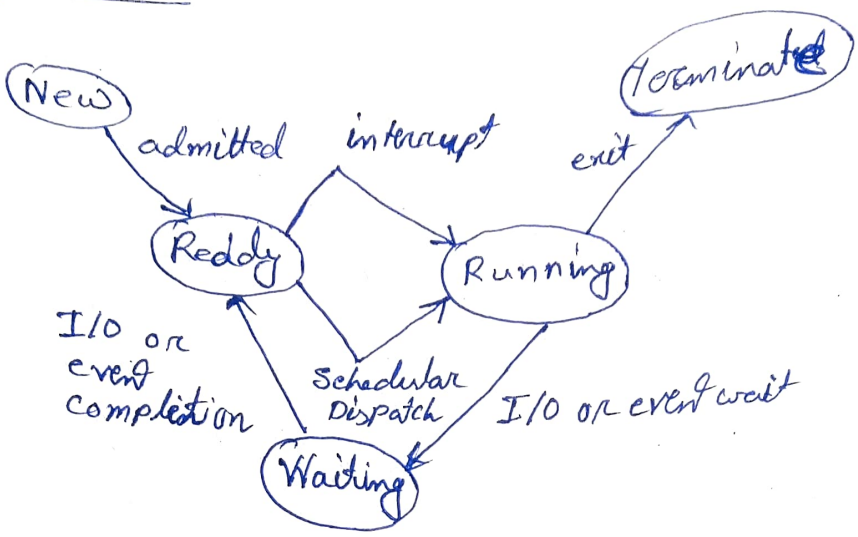


5. What is process? What are the state of a process and describe it with suitable picture:

A: A process is the instance of a computer program that is being executed by one or many threads. It contains the program code and its activity.

States:-

- NEW → Process is being created
- READY → Process is waiting to be assigned to a processor.
- RUNNING → Instructions are being executed.
- WAITING → Process is waiting for some event to occur
- Terminated → The process has finished execution.



6. What is process control block(PCB)? (Describe all information it contain)

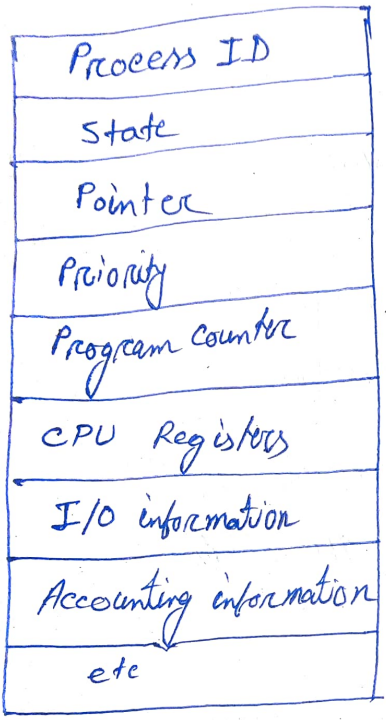
A: Process Control Block :-

It is a data structure which contains -

- Process state: Can be running waiting e.t.c
- Process ID and the parent process ID
- CPU registers and program counter - Program counter holds the address of the next instruction to be executed for that process.
- CPU scheduling information: Such as priority information and pointers to scheduling queues.
- Memory management information : i.e Page tables or segment tables.

• Accounting Information : The user and kernel CPU time consumed, account numbers, limits .etc

• I/O status of information : Devices allocated, open file tables .etc



7. Write difference between Long term scheduler, short term scheduler and Medium term scheduler.

	Long term scheduler	short term scheduler	Medium term scheduler.
1.	It is a Job scheduler	It is a CPU scheduler	It is process swapping scheduler.
2.	It takes process from Job pool	It takes the process from ready state	It takes process from running or dead state.
3.	Speed is lesser than short term	It is fastest among two others	Speed is between the other two.
4.	controls the degree of multiprogramming	Has less control over degree of multiprogramming	Reduces the degree of multiprogramming.

8. What do you mean by context switch with proper example?

A: Switching the CPU to another process requires saving the state of old process and loading the saved state for the new process. The task is known as context switch.

example :- In the linux kernel context switching registers stack pointer program counter flushing the translation lookaside buffer (TLB) and loading the page table of the next process to run.

9. What is Thread? Write a difference between Process and Thread. What are the benefits of Thread?

A: A thread is a flow of execution through the process code with its own program counter that keeps track of which instruction to execute next system register which held it's current working variables and a stack which contains execution history.

Process vs Thread

Process	Thread
1. Process is heavy weight or intensive	1. It is a light weight taking lesser resources than a process.
2. Process switching needs interaction with OS	2. Thread switching does not need to interact with OS.

Process	Thread
3. In multiple processing environments each process execute the same code but has its own memory and file resources	3. All thread can share same set of open files child process.

⇒ Benefits of Thread —

1) minimize the context switching.
2) provides concurrency within a process
3) Efficient communication.
4) Allows utilization of multiprocessor architechtures to a greater scale and efficiency.

10. What is User Level Threads and Kernel Level Threads, Write the difference between User level Thread and Kernel level Thread.

A: User level thread :- The thread management is not aware of the existance of threads. The thread library contains code for creating and destroying threads for passing message and data between threads for scheduling thread execution and for saving and restoring thread contexts.

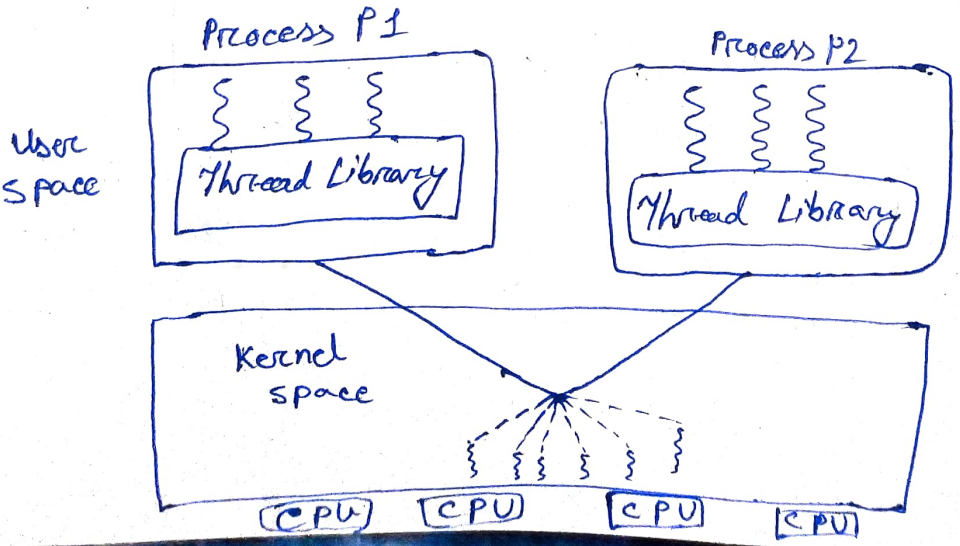
Kernel level thread :- Thread management is done by the Kernel. There is no thread management code in application area. Kernel 4 threads are supported directly by the OS. Any application can be programmed to be multithreaded

- Difference

User level thread	Kernel level thread
1. Are faster to create and manage	1. Threads are slower to create and manage.
2. Implementation is by thread library	2. OS supports creation of Kernel threads.
3. It's generic and can run on any device	3. It is specific to the OS
4. Multithreaded application cannot take advantage of multiprogramming	4. Kernel routines themselves can be multithreaded.

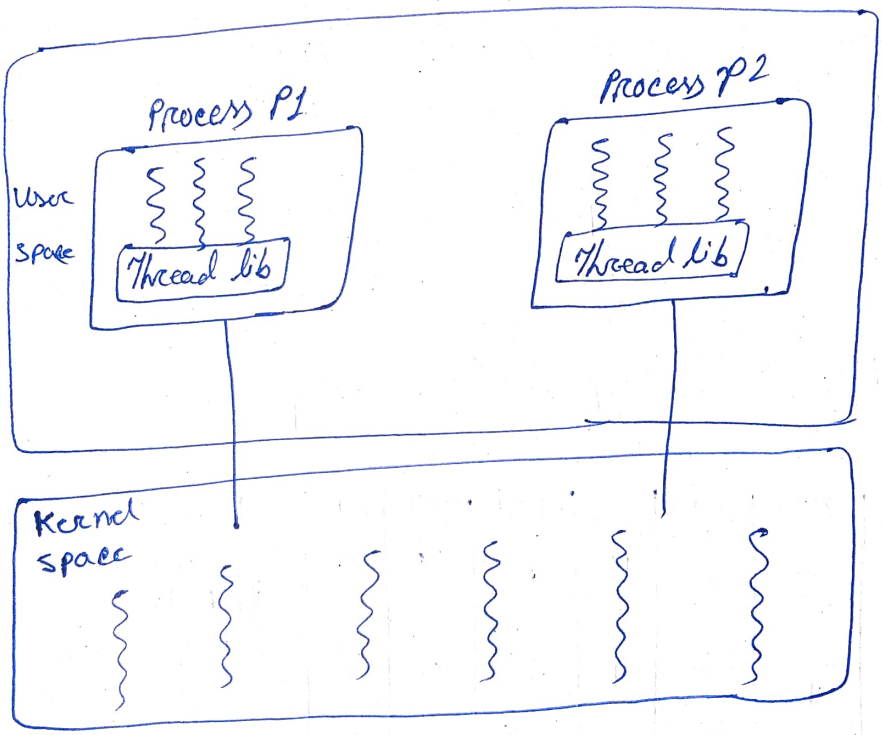
11. i) Many to many Model —

The many to many model multiplexes any number of user threads onto an equal or similar number of Kernel threads.
many to many model of 6 user level threads multiplexing with 6 Kernel level threads.



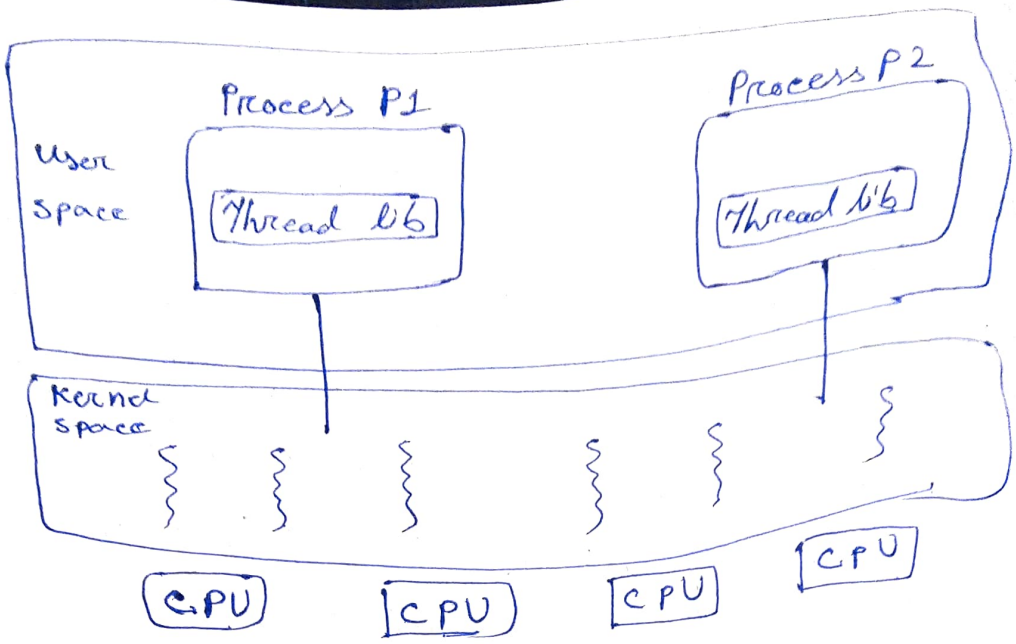
Many to one model –

Many to one model maps many user level threads to one kernel-level thread. Thread management is done in user space by the thread library. When thread makes a blocking system call. the entire process will be blocked only one thread can access the kernel at a time, so multiple threads are unable to run in a parallel on multiprocessors.



One to one model :-

There is one to one relationship of user-level thread to the kernel level thread. This model provides more concurrency than the money to one model. It also allows another ~~model~~ thread to run when thread makes a blocking system call. It supports multiple threads to execute in parallel on microprocessor.

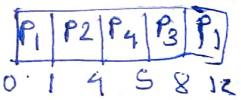


12.

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	3
P4	4	1

Process	AT	BT	CT	TAT	WT	RT
P1	0	5	12	12	7	0
P2	1	3	4	3	0	0
P3	2	3	8	6	3	8
P4	4	1	5	1	0	0

Gnatt chart



Avg TAT $(12+3+6+1)/4$

$= 5.5$

Avg Waiting Time =

$(7+0+3+0)/4$

$= 2.5$

Avg Response Time =

$(0+0+3+0)/4$

$= 0.75$

13.

Process	Burst Time
P1	24
P2	3
P3	3

Gantt chart:

P1	P2	P3	P1

0 4 7 10 30

Process	BT	CT	WT
P1	24	30	6
P2	3	7	4
P3	7	10	7

Avg. waiting Time =

17/3

= 5.67

14.

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

Gantt chart

P2	P5	P1	P3	P4

0 1 6 16 18 19

Avg waiting time = $(1+6+16+18)/5 + 0 = 8.2$

15. What is semaphore? What are the operations on it? Determine the solution of 'Readers-Writers problem' using semaphore?

A:- A semaphore is an integer variable that is used to solve the critical section problem by using two atomic operation wait() and signal().

Wait()	signal()
This operation is used to acquire	This operation is used to release the lock

wait(s){
while(s<=0);
s--;
}
	signal(s){
	++s;
	}

A.

- **solution of Readers Writers Problem**

```
shared data structures;
semaphore rw_mutex = 1;
semaphore mutex = 1
int read_count = 0;
```

Writer code:

```
while (true){
    wait (rw_mutex);
    /* writing is performed */
    signal (rw_mutex);
}
```

Reader code:

```
while(true){
    wait(mutex);
    read_count++;
    if(read_count==1)
        wait(rw_mutex);
    signal(mutex);
/* reading is performed */
    wait(mutex);
    read_count--;
    if(read_count==0)
        signal(rw_mutex);
    signal(mutex);
}
```

16. Determine the slow solution of 'Dining-Philosopher's Problem' using semaphore.

A:

```
semaphore chopstick[5]
while(true){
    wait(chopstick[i]);
    wait(chopstick[(i+1)%5]);
/* eat for a while */
    signal(chopstick[i]);
    signal(chopstick[(i+1)%5]);
```

To avoid deadlock, do either one of the following;

1) Allow at most four philosophers;
2) Allow a particular to pick up both chopsticks or none.

3) Use asymmetric solution odd numbered.

~~Phil~~ philosopher picks left chopstic then right and even numbered philosopher picks right chopstic then left.

~~17. What do you mean by deadlock?~~

17. Producer consumer Problem using Semaphore

```
Semaphore empty = 1;
Semaphore full = 1;
Semaphore mutex = 1;
```

Producer:

```
while(true){
    wait(empty);
    wait(mutex);
/* producer an item to buffer */
    signal(mutex);
    signal(full);
}
```

consumer:

```
while(true){
    wait(full);
    wait(mutex);
/* consume an item from buffer */
    signal(mutex);
    signal(empty);
}
```